9.0 EXAMPLES AND SAMPLE PROBLEMS

The following describes execution of the FEHM code. Section 9.1 discusses the construction of an input file. Section 9.2 illustrates the entire procedure for executing the FEHM code using terminal input. Example 1 describes the setup and results from a simple 2-D heat-conduction simulation. The remaining sections provide more complex example problems and deal only with problem setup and expected results.

9.1 Constructing an Input File

FEHM is a very general simulation code. Thus it is preferable to discuss the construction of an input file from a problem-oriented point of view. In what follows, the needs of the physical problem (initial conditions, boundary conditions, etc.) will be addressed in terms of the macro statements.

Initial conditions. These are needed for every problem, even if it is a steady-state simulation. If the simulation is comprised of fully saturated water flow or heat conduction only, then the appropriate control statement would be **init** (page 37). The use of **init** also allows the specification of initial temperature and pressure (gravity) gradients. If two-phase flow is prescribed (thermal or isothermal), then entering the initial conditions through the control statement **pres** (page 44) is more convenient. Initial values for noncondensible gas are handled in the **ngas** (page 40) control statement. It should be remembered that if a restart file is present, those values will have precedence over values input in control statement **init** but not over values input in control statement **pres**. Solute initial conditions are prescribed through the control statment **trac** (page 61).

Boundary conditions. Fluid and heat flow boundary conditions can be prescribed through control statements **pres**, **flow** (page 34), **hflx** (page 37), and **rflx** (page 50). Boundary conditions are entered with **pres** by specifying a negative phase-state designation (the code will actually use the absolute value of the phase-state designation). In this case, the code will keep the variable values constant at whatever value was prescribed in **pres**. Flowing pressures are input with the **flow** control statement. Solute boundary conditions are prescribed through the control statment **trac**.

Material- and energy-balance equations. The choice of the coupled system equations is made in control statements **sol** (page 59), **ngas**, and **airwater** (page 22).

Rock or media properties. These are found in the **rock** (page 53) and **perm** (page 43) control statements.

Fluid properties. These are found in control statement **eos** (page 32), which is optional. If **eos** is not invoked, then the properties of water and air included in the code are used. Relative permeabilities, depending on both the fluid and media type, are found in control statement **rlp** (page 50).

Mesh geometry and nodal coordinates. This geometry information is found in control statements **coor** (page 26) and **elem** (page 30). This information is usually created with a mesh-generation program.

Simulation time. The time-stepping information, including printout intervals and time-step sizing, is found in control statement **time** (page 60).

Numerics. Convergence criteria, upwinding parameters, fill-in for the preconditioned conjugate gradient solver and geometry type (2-D, 3-D, radial) are entered with control statement **ctrl** (page 27).

Advanced iteration control. Reduced-degree-of-freedom methods are invoked with the **iter** (page 38) control statement. One important quantity entered with this statement is the maximum time for the job to run on the computer.

Sources and sinks. These are input with the control statement **flow**. Care must be taken as the parameters have different meanings for different physical models.

The following table lists the input macros that should be used to formulate various types of problems.

Table VII. Required and optional macros by problem type

Problem type : Heat conduction		**Problem type : Water/water vapor/heat** Equivalent continuum, dual porosity*, dual permeability**	
Required macros	**Optional macros**	**Required macros**	**Optional macros**
title	alti	title	alti
cond	cont	cond	cap
coor	finv	coor	cont
ctrl	flo2	ctrl	eos
elem	flxo	elem	finv
flow or hflx	iter	init or pres	flow
init or pres	node or nod2	perm	flo2
rock	renm	rlp	hfxo
sol	rflx	rock	hflx
time	stea	sol	iter
stop	text or comments (#)	time	node or nod2
	user	stop	ppor
	vcon		pres
	zone	dual (* only)	renm
		dpdp (** only)	rflx
			rxn
			stea
			text or comments (#)
			trac
			vcon
			velo
			zone

Table VII. Required and optional macros by problem type (continued)

Problem type : Air/water/water vapor/heat or gas/water/NAPL/heat Equivalent continuum, dual porosity* dual permeability**		**Problem type : Air/water/no heat** Equivalent continuum, dual porosity* dual permeability**	
Required macros	**Optional macros**	**Required macros**	**Optional macros**
title	comments (#)	title	comments (#)
cond	adif	airwater	alti
coor	alti	coor	cap
ctrl	cap	ctrl	cond
elem	cont	elem	cont
flow or hflx	eos	flow or hflx	eos
init or pres	finv	init or pres	finv
ngas	flow	node or nod2	flow
perm	flo2	perm	flo2
rlp	flxo	rock	flxo
rock	hflx	sol	hflx
sol	iter	time	iter
time	node or nod2	stop	ppor
stop	ppor		pres
	renm	dual (*only)	renm
dual (*only)	rflx	dpdp (**only)	rflx
dpdp (**only)	rxn		rxn
	stea		stea
	text		text
	trac		trac
	vcon		vcon
	velo		velo
	zone		zone

9.2 Code Execution

To run FEHM, the program executable file name is entered at the system prompt:

<PROMPT> xfehmn

The I/O file information is provided to the code from an input control file or the terminal. The default control file name is *fehmn.files*. If a control file with the default name is present in the directory from which the code is being executed, no terminal input is required. If the default control file is not present, input prompts are written to the screen. A short description of the I/O files used by FEHM precedes the initial prompt. The following assumes the default control file was not found in the execution directory (for this example /home/fehm/heat2d).

After the command **xfehmn** is given, the code queries the user regarding the input files, as follows:

Enter name for iocntl -- default file name: not using

[(name/na or not using), RETURN = DEFAULT]

This query asks for a control file name. If a control file name is entered, no further terminal input is required. Figure 3 shows the control file that would produce the same results as the terminal responses discussed below and illustrated in Fig. 4.

```
/home/fehm/heat2d/input/heat2d.in
/home/fehm/heat2d/input/heat2d.in
/home/fehm/heat2d/input/heat2d.in
/home/fehm/heat2d/output/heat2d.out

/home/fehm/heat2d/output/heat2d.fin
/home/fehm/heat2d/output/heat2d.his




/home/fehm/heat2d/output/heat2d.chk
some
0
```

Figure 3. Input control file for heat-conduction example.

Files that are not needed for output can be represented with a blank line. If names are not provided for the write file or the data check file, the code will use the following defaults: *fehmn.fin* and *fehmn.chk*. Following the file names is the flag that controls terminal output. The last line of the file is the user subroutine number. Omitting these values results in no terminal output and no user subroutine call. For now, we assume a carriage return <cr> is entered and a control file is not being used. The following query will appear

Enter name for inpt -- default file name: fehmn.dat

[(name/na or not using), RETURN = DEFAULT]

This query asks for an input file name. If a <cr> is given, the default *fehmn.dat* is used for the input file. We shall assume that the input file name entered is

input/heat2d.in

Note that a subdirectory containing the file is also given. If the file did not exist, the code would repeat the prompt for an input file. Next, the code would query to determine if the prefix of the input file name (the portion of the name preceding the final "." or first space) should be used for code-generated file names.

Do you want all file names of the form input/heat2d.* ? [(y/n), RETURN = y]
*** Note: If "y" incoor and inzone will equal inpt ***

A <cr> will produce files with identical prefixes, including the subdirectory. If the response is negative, the code will query for the names of all required files. Assume we enter "n".

Enter name for incoor -- default file name: input/heat2d.in

[(name/na or not using), RETURN = DEFAULT]

(See Fig. 4 for the remaining file name queries.)

Next a query for terminal output appears.

tty output -- show all reference nodes, selected reference nodes, or none:
[(all/some/none), RETURN = none]

An "all" reply prints out the primary-node information to the terminal at every time step. A "some" reply prints a selected subset of the node information. A reply of "none" suppresses all tty output with the exception of error messages printed if code execution is terminated abnormally or when the maximum number of iterations are exceeded. Assume we enter "some".

The next query concerns the subroutine USER. This subroutine is used for special purposes and is not available to the general user.

user subroutine number (provided to subroutine USER before every time step):
[RETURN = none]

Assume a <cr> is entered.

The code will then print a summary of the I/O files to be used.

The final query regards the acceptance of the file set just created. A "yes" reply denotes that the user has accepted the file set and the code proceeds with calculations. A "no" reply starts the query sequence again so I/O file names may be reentered or modified. A "stop" reply stops the current computer job.

If data is OK enter yes to continue, no to restart terminal input,
or stop to end program: [(yes/no/stop), RETURN = yes]

Screen output for this example execution using terminal input is shown in Fig. 4. User responses are shown in *italics*.

<PROMPT> *xfehmn*

Version FEHMN XX-XX-XX 94/01/11 09:24:04

**** Default names for I/O files ****

control file	: fehmn.files
input file	: filen.*
geometry data file	: filen.*
zone data file	: filen.*
output file	: filen.out
read file (if it exists)	: filen.ini
write file (if it exists)	: filen.fin
history plot file	: filen.his
tracer history plot file	: filen.trc
contour plot file	: filen.con
dual or dpdp contour plot file	: filen.dp
stiffness matrix data read/write file	: filen.stor
input check file	: filen.chk

**** where ****
"filen.*" may be 100 characters maximum. If a name is not entered when prompted for, a default file name is used. "fehmn.dat" is the default used for the input file name.

**** note ****
A save file and input check file are always written. If you do not provide a name for these files, the following defaults will be used: fehmn.fin, fehmn.chk

Enter name for iocntl -- default file name: not using

[(name/na or not using), RETURN = DEFAULT]
<cr>

Enter name for inpt -- default file name: fehmn.dat

[(name/na or not using), RETURN = DEFAULT]
input/heat2d.in

Do you want all file names of the form example/heat2d.* ? [(y/n), RETURN = y]
*** Note: If "y" incoor and inzone will equal inpt ***
n

Enter name for incoor -- default file name: input/heat2d.in

[(name/na or not using), RETURN = DEFAULT]
<cr>

Figure 4. Terminal query for FEHM example run .

Enter name for inzone -- default file name: input/heat2d.in

[(name/na or not using), RETURN = DEFAULT]
<cr>

Enter name for iout -- default file name: input/heat2d.out

[(name/na or not using), RETURN = DEFAULT]
output/heat2d.out

Enter name for iread -- default file name: input/heat2d.ini

[(name/na or not using), RETURN = DEFAULT]
na

Enter name for isave -- default file name: input/heat2d.fin

[(name/na or not using), RETURN = DEFAULT]
output/heat2d.fin

Enter name for ishis -- default file name: input/heat2d.his

[(name/na or not using), RETURN = DEFAULT]
output/heat2d.his

Enter name for istrc -- default file name: input/heat2d.trc

[(name/na or not using), RETURN = DEFAULT]
na

Enter name for iscon -- default file name: input/heat2d.con

[(name/na or not using), RETURN = DEFAULT]
na

Enter name for iscon1 -- default file name: input/heat2d.dp

[(name/na or not using), RETURN = DEFAULT]
na

Enter name for isstor -- default file name: input/heat2d.stor
[(name/na or not using), RETURN = DEFAULT]
na

Enter name for ischk -- default file name: input/heat2d.chk

[(name/na or not using), RETURN = DEFAULT]
output/heat2d.chk

Figure 4. Terminal query for FEHM example run (continued).

tty output -- show all reference nodes, selected reference nodes, or none:
[(all/some/none), RETURN = none]
some

user subroutine number (provided to subroutine USER before every time step):
[RETURN = none]
<cr>

First reference output node will be written to tty

File purpose - Variable - Unit number - File name

control	- iocntl	- 0	- not using
input	- inpt	- 11	- input/heat2d.in
geometry	- incoor	- 11	- input/heat2d.in
zone	- inzone	- 11	- input/heat2d.in
output	- iout	- 14	- output/heat2d.out
initial state	- iread	- 0	- not using
final state	- isave	- 16	- output/heat2d.fin
time history	- ishis	- 17	- output/heat2d.his
time his.(tr)	- istrc	- 18	- not using
contour plot	- iscon	- 19	- not using
con plot (dp)	- iscon1	- 20	- not using
fe coef stor	- isstor	- 21	- not using
input check	- ischk	- 22	- output/heat2d.chk

Value provided to subroutine user: not using

If data is OK enter yes to continue, no to restart terminal input,
or stop to end program: [(yes/no/stop), RETURN = yes]
<cr>

Figure 4. Terminal query for FEHM example run (continued).